

# ПРОГРАММНЫЙ КОМПЛЕКС SYSTEME PLATFORM

## SePlatform.HMI.Dashboards 0.1

### Описание компонентов

Редакция	1
Соответствует версии ПО	1.0.0



© ООО «СИСТЭМ СОФТ», 2022-2023. Все права защищены.

Авторские права на данный документ принадлежат ООО «СИСТЭМ СОФТ». Копирование, перепечатка и публикация любой части или всего документа не допускается без письменного разрешения правообладателя.

# Содержание

1. Руководство пользователя.....	4
1.1. О продукте.....	4
1.2. Подготовка к работе.....	4
1.2.1. Требования к окружению.....	4
1.2.2. Установка, удаление или восстановление .....	4
1.3. Встраивание SePlatform.HMI.Dashboards в проекты HMI.....	4
1.4. Конфигурирование источника библиотеки SePlatform.HMI.Dashboards.....	4
2. Описание компонентов библиотеки .....	6
2.1. ComboChart (Комбинированная диаграмма).....	7
2.2. PieChart_1 (Круговая диаграмма).....	8
2.3. PieChart_2 (Круговая диаграмма).....	10
2.4. StackedBarChart (Диаграмма с накоплением) .....	12
2.5. ProgressBar (Индикатор выполнения).....	14
2.6. StackedColumnChart (Столбчатая диаграмма с накоплением) .....	15
2.7. PieIndicator_1 (Круговой индикатор) .....	16
2.8. PieIndicator_2 (Круговой индикатор) .....	17
2.9. HistogramGrouping (Гистограмма с группировкой).....	18
2.10. RPM_1 (Диаграмма спидометр).....	20
2.11. RPM_2 (Диаграмма спидометр).....	21
2.12. RPM_3 (Диаграмма спидометр SA) .....	23
2.13. RPM_4 (Диаграмма спидометр SA) .....	24
2.14. CircularArea_1 (Лепестковая диаграмма) .....	26
2.15. CircularArea_2 (Лепестковая диаграмма) .....	28
2.16. HorizontalBarPrc (Индикатор прогресса).....	29
2.17. VerticalBar_1 (Вертикальный индикатор прогресса).....	30
2.18. VerticalBar_2 (Вертикальный индикатор прогресса).....	31
2.19. VerticalBar_3 (Вертикальный индикатор прогресса).....	32
2.20. VerticalBar_4 (Вертикальный индикатор прогресса).....	34
2.21. LineChart (Линейный график).....	35
2.22. BarChart (Столбчатая диаграмма).....	36
2.23. LevelChart_1 (Горизонтальный уровень) .....	38
2.24. LevelChart_2 (Вертикальный уровень).....	39

## 1. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

### 1.1. О продукте

SePlatform.HMI.Dashboards – набор графических компонентов на базе SePlatform.HMI, предназначенный для анализа и сбора данных в режиме реального времени.

Используйте SePlatform.HMI.Dashboards совместно с SePlatform.HMI. Подробнее в п. 1.3. Подключение библиотеки к проекту.

### 1.2. Подготовка к работе

#### 1.2.1. Требования к окружению

Для работы SePlatform.HMI.Dashboards должны быть установлены:

- **SePlatform.HMI** – среда разработки проектов автоматизации. Рекомендуется использование версии HMI не ниже 2.0.\*;

Если приложение используется в веб-версии проекта автоматизации, то установите дополнительно:

- **SePlatform.HMI.WebViewer** – для просмотра проектов SePlatform.HMI в веб-интерфейсе;

#### 1.2.2. Установка, удаление или восстановление

##### ОС Windows

Для установки, удаления или восстановления SePlatform.HMI.Dashboards запустите установочный файл SePlatform.HMI.Dashboards \_<version>.msi и следуйте инструкциям мастера.



#### ПРИМЕЧАНИЕ

В названии файла, <version> – номер версии компонента,

### 1.3. Встраивание SePlatform.HMI.Dashboards в проекты HMI

Чтобы встроить SePlatform.HMI.Dashboards в проект автоматизации, разработанный в SePlatform.HMI, подключите проект расширения в свой проект в виде внешнего модуля.

Встраиваемые компоненты библиотеки расположены по следующему пути:

C:\Program Files\SePlatform\SePlatform.HMI.Extensions\Dashboards

### 1.4. Конфигурирование источника библиотеки SePlatform.HMI.Dashboards

В библиотеку Dashboards встроен источник Global\_Main, к которому подключаются все элементы. Для конфигурирования источника необходимо в обработчике (выполнить код) обратиться к источнику и сконфигурировать необходимое свойство. По умолчанию:

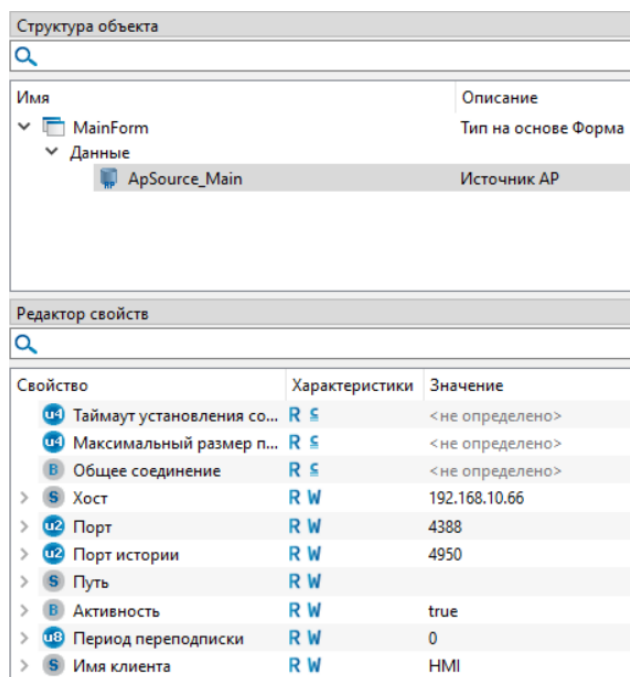
IP-адрес источника: 127.0.0.1;

Порт: 4388;

Порт истории: 4950

Ниже представлен один из вариантов настройки источника, когда нужно задать источнику библиотеки Global\_Main IP-адрес, указанный в некотором другом источнике ApSource\_Main.

На форме MainForm создан источник ApSource\_Main с некоторыми базовыми настройками.



Форма MainForm является главной, поэтому по открытии этой формы должен выполняться обработчик с действием «Выполнить код». К свойству «хост», которое требуется сконфигурировать, можно обратиться по тегу Location. На рисунке ниже представлен код присвоения IP-адреса источника ApSource\_Main источнику библиотеки Global\_Main.

```
Global_Main.Location = ApSource_Main.Location;
```







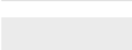
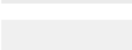

Примечание: иногда нужно обратиться к библиотеке напрямую, поэтому код будет выглядеть следующим образом:



```
Dashboards.Global_Main.Location = ApSource_Main.Location;
```










После запуска формы в Runtime источнику библиотеки будет присвоен необходимый IP-адрес, элементы библиотеки подключатся к серверу и заберут необходимые данные.



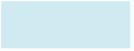


## 2. ОПИСАНИЕ КОМПОНЕНТОВ БИБЛИОТЕКИ

Библиотека содержит в себе настроенные стили и компоненты. И то и другое используется для ускорения работы и быстрого внесения изменений по всему проекту. Компоненты библиотеки поддерживают изменение цветов элементов, из которых они состоят. Доступная цветовая палитра приведена ниже.

	VeryDarkGray
	DarkGray
	SonicSilver
	MediumGray
	Silver
	LightGray
	VeryLightGray
	ChryslerWhite
	White

	LapisLazuli
	YaleBlue
	LightSteelBlue
	AliceBlue
	CeruleanCrayola
	MiddleBlueGreen
	ColumbiaBlue
	JungleGreen
	Straw
	CandyPink
	LightPink
	MistyRose
	SelectiveYellow

	RoyalPurple
	MiddleBluePurple
	PurpleMountainMajesty
	GlossyGrape
	GrayishViolet
	Thistle
	LanguidLavender
	LightGrayishViolet
	LavenderWeb
	GhostWhite

	GOST_Water
	GOST_Steam
	GOST_Air
	GOST_CombustibleGases
	GOST_NonFlammableGasesAndLiquids
	GOST_Acid
	GOST_Alkali
	GOST_CombustibleLiquids
	GOST_Oxygen

Для выбора цвета в соответствующем поле нужно указать, через "Сослаться": **unit.Colors.\**нужное наименование*\***

Например: unit.Colors.YaleBlue

В шаблонах библиотеки преимущественно используется тип данных float4. Если тип данных не совпадает, требуется произвести конвертирование типов. Как перевести в различные форматы описано в документации: **SePlatform.Om** → **Типы данных** → **Явное преобразование элементарных типов**.

Для изменения размера используется коэффициент масштабирования.

Обозначение в документе:

**SA** – Situational Awareness – Ситуационное восприятие.

## 2.1. ComboChart (Комбинированная диаграмма)

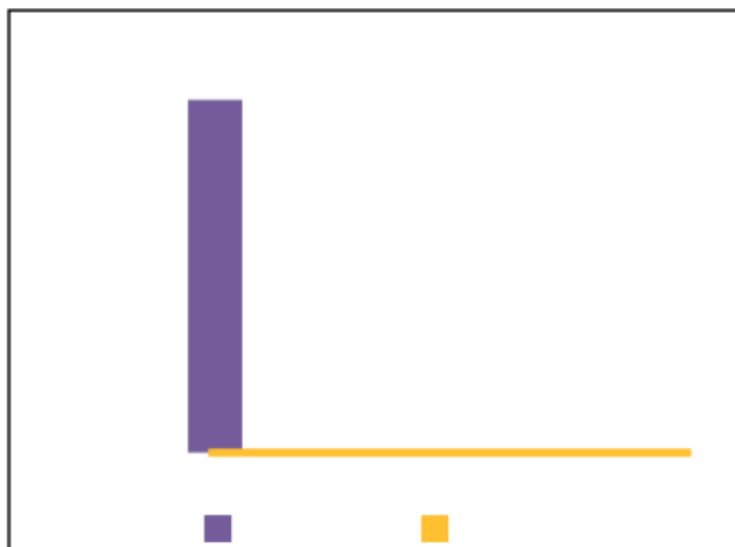
Комбинированная диаграмма – это комбинация двух диаграмм: столбчатой диаграммы с линейной. На оси X доступно 6 рядов. Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмента шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 8 свойств.

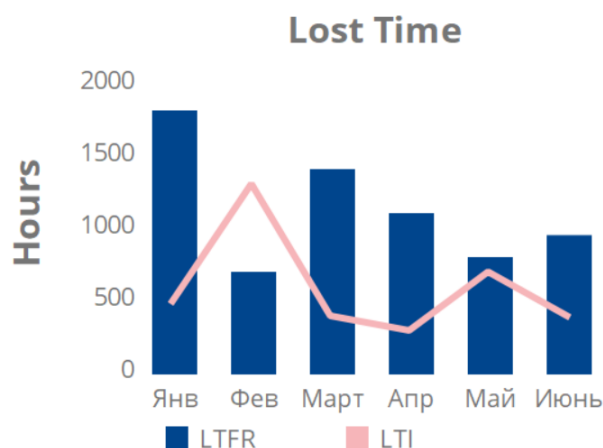
- S Путь до объекта
- S Название диаграммы
- S Подпись оси Y
- S Легенда столбца
- S Легенда линии
- u4 Цвет столбцов
- u4 Цвет линии
- u4 Цвет шрифта

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.ComboChart

#### f4 Object.ComboChart.Column\_1

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

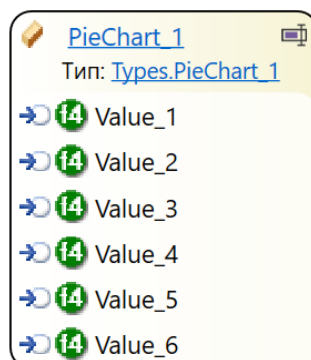


## 2.2. PieChart\_1 (Круговая диаграмма)

Круговая диаграмма – круговая динамическая диаграмма, разделенная на 6 срезов для иллюстрации числовой пропорции. Доступно отображение 1, 2, 3, 4, 5 или 6-ти срезов с процентным соотношением.

Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:

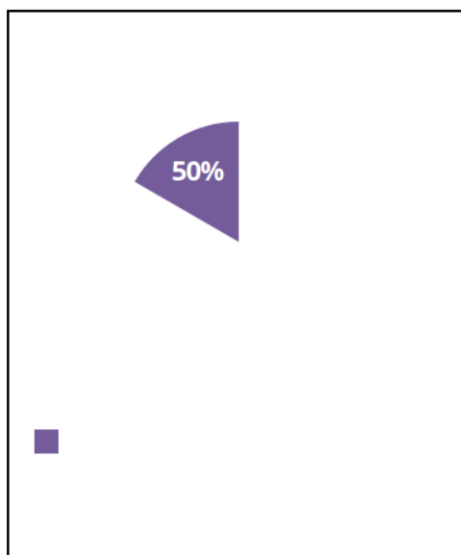




На вход подается значение некоторых величин (до 6-ти). Процентное соотношение вычисляется автоматически.

На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 17 свойств.

- S Путь до объекта
- i1 Число переменных
- S Название диаграммы
- S Легенда Части 1
- S Легенда Части 2
- S Легенда Части 3
- S Легенда Части 4
- S Легенда Части 5
- S Легенда Части 6
- u4 Цвет Части 1
- u4 Цвет Части 2
- u4 Цвет Части 3
- u4 Цвет Части 4
- u4 Цвет Части 5
- u4 Цвет Части 6
- u4 Цвет шрифта внутри диаграммы
- u4 Цвет шрифта за диаграммой

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

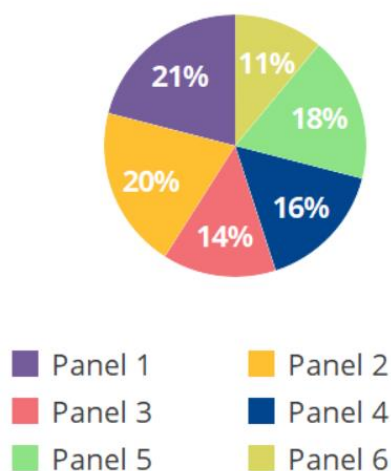
В данном примере путь до объекта: Object.PieChart\_1

f4 Object.PieChart\_1.Value\_1

**Число переменных.** Указывается число переменных для отображения на диаграмме. Диапазон значений: 1–6.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

### Lighting Zone Ranking

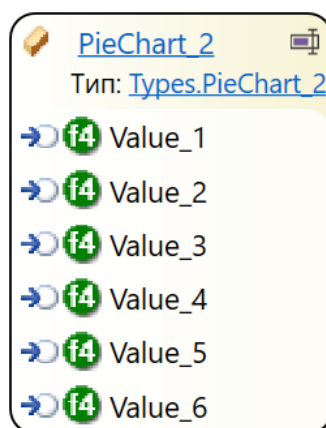


## 2.3. PieChart\_2 (Круговая диаграмма)

Круговая диаграмма – круговая динамическая диаграмма, разделенная на 6 частей для иллюстрации числовой пропорции. Доступно отображение 1, 2, 3, 4, 5 или 6-ти срезов с процентным соотношением.

Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:



На вход подается значение некоторых величин (до 6). Процентное соотношение вычисляется автоматически.

На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки. Для конфигурации диаграммы необходимо указать 17 свойств.

- S Путь до объекта
- i1 Число переменных
- S Название диаграммы
- S Легенда Части 1
- S Легенда Части 2
- S Легенда Части 3
- S Легенда Части 4
- S Легенда Части 5
- S Легенда Части 6
- u4 Цвет Части 1
- u4 Цвет Части 2
- u4 Цвет Части 3
- u4 Цвет Части 4
- u4 Цвет Части 5
- u4 Цвет Части 6
- u4 Цвет шрифта внутри диаграммы
- u4 Цвет шрифта за диаграммой

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

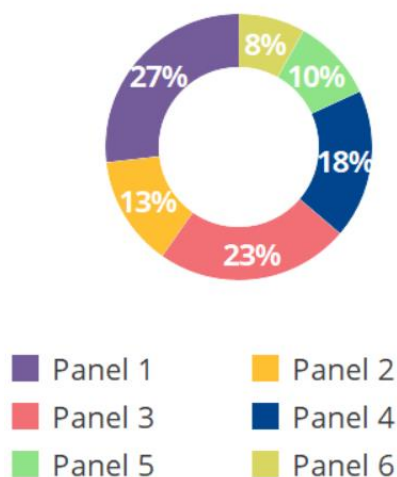
В данном примере путь до объекта: Object.PieChart\_2

i4 Object.PieChart\_2.Value\_1

**Число переменных.** Указывается число переменных для отображения на диаграмме. Диапазон значений: 1–6.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

## Lighting Zone Ranking

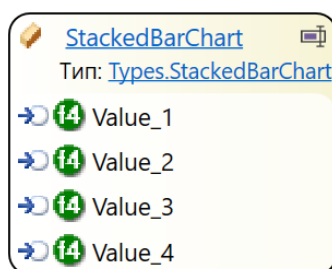


## 2.4. StackedBarChart (Диаграмма с накоплением)

Диаграмма с накоплением показывает соотношение внутренних сегментов в итоговой величине. При этом сегменты располагаются друг за другом — началом для каждого следующего сегмента будет крайняя граница предыдущего. Для элемента доступно отображение 1, 2, 3 или 4-х срезов с процентным соотношением.

Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 13 свойств.

- S Путь до объекта
- i1 Число переменных
- S Название диаграммы
- S Легенда 1
- S Легенда 2
- S Легенда 3
- S Легенда 4
- u4 Цвет среза 1
- u4 Цвет среза 2
- u4 Цвет среза 3
- u4 Цвет среза 4
- u4 Цвет шрифта внутри диаграммы
- u4 Цвет шрифта за диаграммой

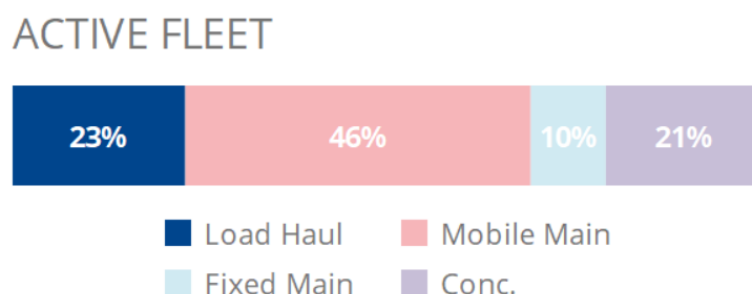
**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.StackedBarChart

i4 Object.StackedBarChart.Value\_1

**Число переменных.** Указывается число переменных для отображения на диаграмме. Диапазон значений: 1–4.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

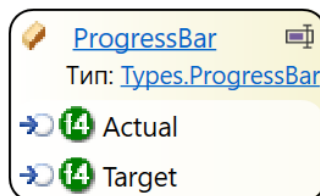


## 2.5. ProgressBar (Индикатор выполнения)

Индикатор выполнения — «заполняется» областью другого цвета по мере выполнения какой-либо задачи, например загрузки файла. Стандартный индикатор процесса. Заполняется слева направо.

Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

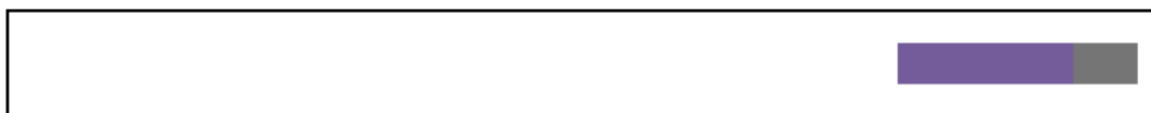
Изображение шаблона в DevStudio:



На вход подается текущее и целевое значения некоторой величины. Число, обозначающее текущее значение записывается в переменную Actual, целевое в Target. Процентное соотношение вычисляется автоматически.





На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:




Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 4 свойства.

-  Путь до объекта
-  Название величины
-  Цвет заливки
-  Цвет шрифта за диаграммой

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.ProgressBar

 Object.ProgressBar.Actual

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

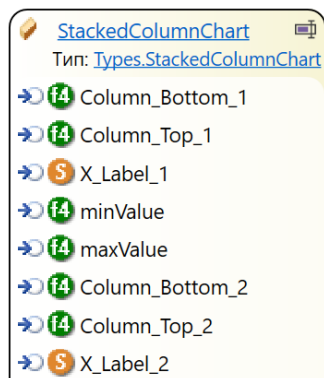


## 2.6. StackedColumnChart (Столбчатая диаграмма с накоплением)

Столбчатая диаграмма с накоплением использует столбцы, разделенные на несколько вложенных столбцов, для визуализации значений нескольких переменных одновременно.

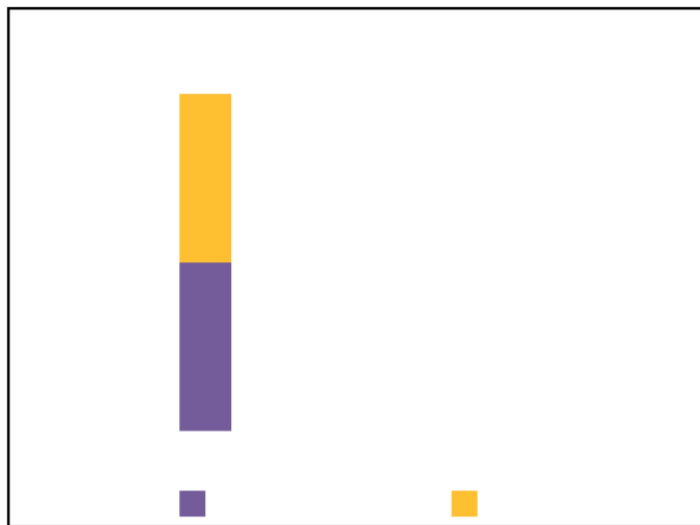
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмента шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 8 свойств.

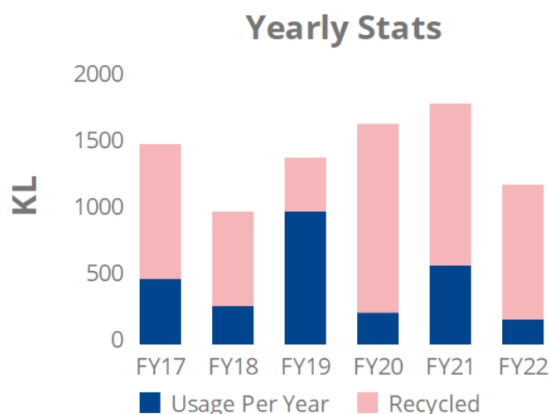
- S Путь до объекта
- S Наименование диаграммы
- S Подпись оси Y
- S Легенда нижних столбцов
- S Легенда верхних столбцов
- u4 Цвет нижних столбцов
- u4 Цвет верхних столбцов
- u4 Цвет шрифта

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.StackedColumnChart

#### Object.StackedColumnChart.Column\_Top\_1

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

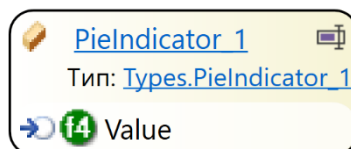


## 2.7. PieIndicator\_1 (Круговой индикатор)

Круговой индикатор — графический элемент, показывающий статус процесса с помощью заполнения окружности по мере его выполнения.

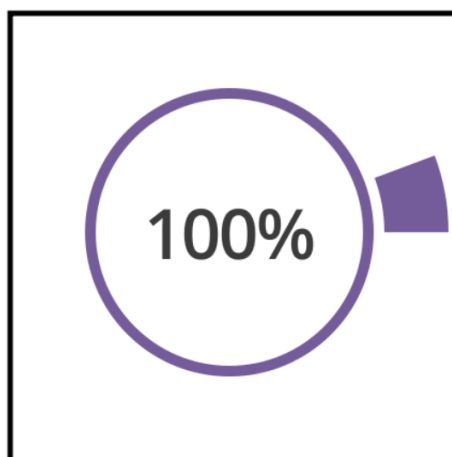
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:



На вход переменной нужно подать некоторое значение в процентах (0-100). Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.



Для конфигурации диаграммы необходимо указать 3 свойства.

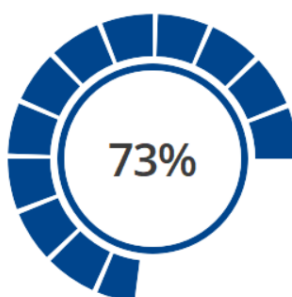
- S Путь до объекта
- u4 Цвет индикатора
- u4 Цвет шрифта

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.PielIndicator\_1

f4 Object.PielIndicator\_1.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

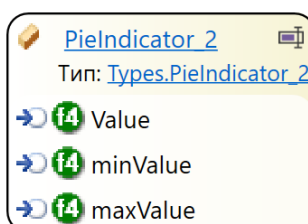


## 2.8. PieIndicator\_2 (Круговой индикатор)

Круговой индикатор — графический элемент, показывающий статус процесса с помощью заполнения окружности по мере его выполнения.

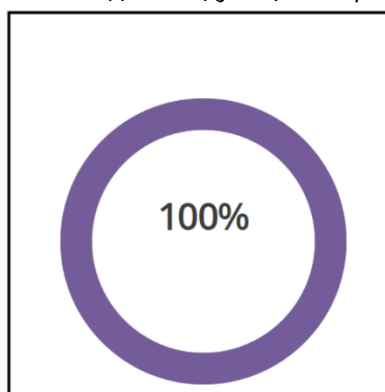
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:









На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:




Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 6 свойств.

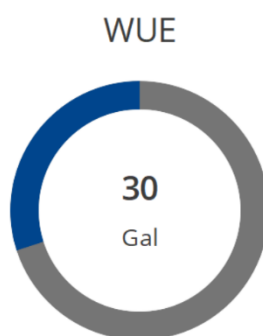
-  Путь до объекта
-  Название диаграммы
-  Ед. изм.
-  Цвет индикатора
-  Цвет фоновой части
-  Цвет шрифта

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.PielIndicator\_2

 Object.PielIndicator\_2.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:



## 2.9. HistogramGrouping (Гистограмма с группировкой)

Гистограмма с группировкой — графическое представление статистических данных, отражающее количество наблюдений в каждой группе. Одна из главных особенностей гистограммы с группировкой заключается в том, что она показывает какое количество наблюдений (число или частоту) попадает в каждую группу.

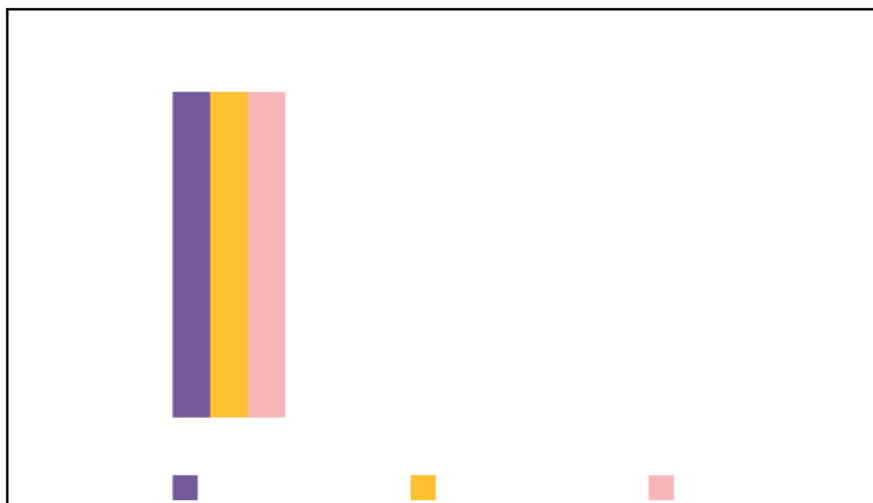
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмента шаблона в DevStudio:














На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 11 свойств.

-  Путь до объекта
-  Число переменных
-  Подпись диаграммы
-  Подпись оси Y
-  Легенда столбца 1
-  Легенда столбца 2
-  Легенда столбца 3
-  Цвет столбца 1
-  Цвет столбца 2
-  Цвет столбца 3
-  Цвет шрифта

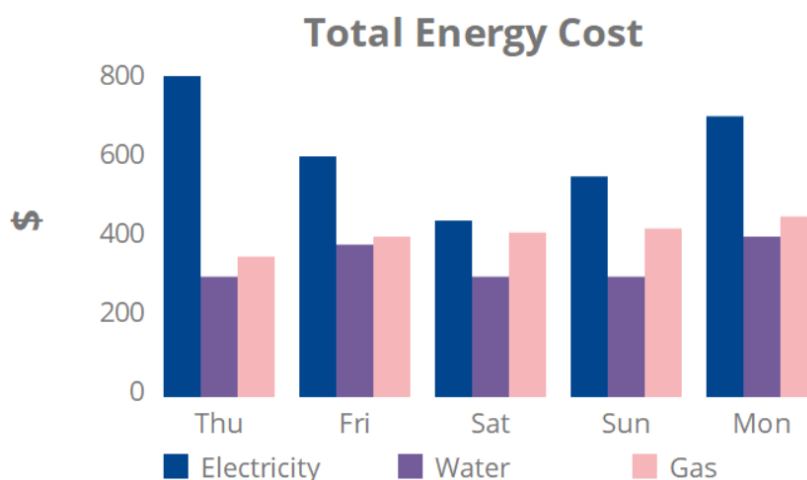
**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.HistogramGrouping

 Object.HistogramGrouping.Column\_1\_1

**Число переменных.** Указывается число столбцов в ряду для отображения на диаграмме. Диапазон значений: 1–3.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

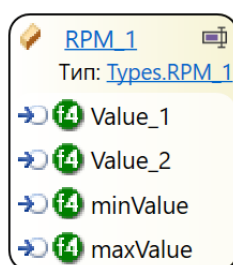


## 2.10. RPM\_1 (Диаграмма спидометр)

Диаграмма спидометр – диаграмма, отображающая уровень показателя согласно заданной шкале.

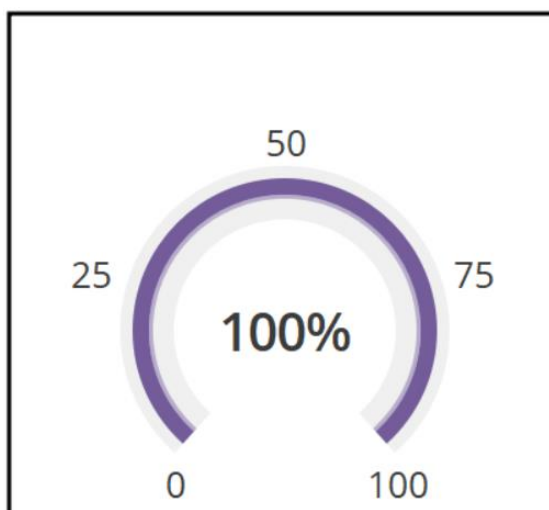
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 8 свойств.

- S** Путь до объекта
- i1** Число переменных
- S** Название диаграммы
- S** Ед. изм.
- u4** Цвет широкой части
- u4** Цвет узкой части
- u4** Цвет фона
- u4** Цвет шрифта

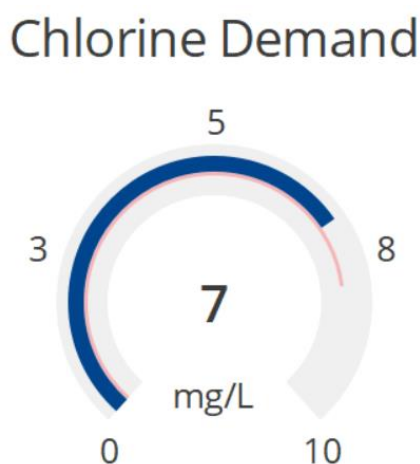
**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.RPM\_1

**f4** Object.RPM\_1.Value\_1

**Число переменных.** Указывается число переменных для отображения на диаграмме. Диапазон значений: 1–2.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

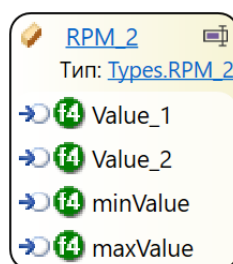


## 2.11. RPM\_2 (Диаграмма спидометр)

Диаграмма спидометр – диаграмма, отображающая уровень показателя согласно заданной шкале.

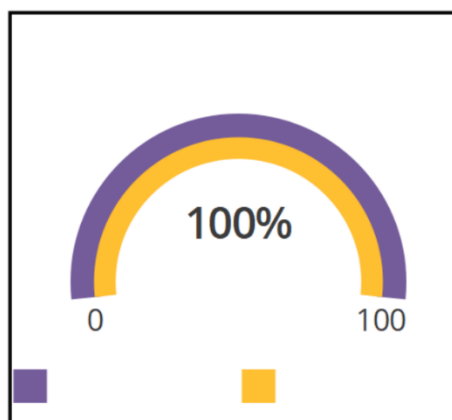
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 9 свойств.

- S Путь до объекта
- i1 Число переменных
- S Название диаграммы
- S Ед. изм.
- S Легенда внешней части
- S Легенда внутренней части
- u4 Цвет внешней части
- u4 Цвет внутренней части
- u4 Цвет шрифта

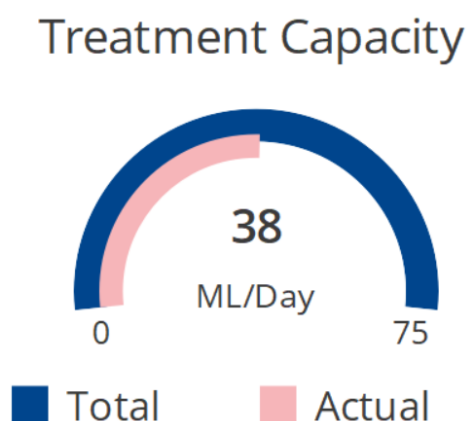
**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.RPM\_2

f4 Object.RPM\_2.Value\_1

**Число переменных.** Указывается число переменных для отображения на диаграмме. Диапазон значений: 1–2.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

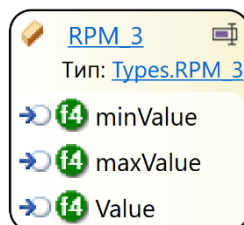


## 2.12. RPM\_3 (Диаграмма спидометр SA)

Диаграмма спидометр – диаграмма, отображающая уровень показателя согласно заданной шкале. Циферблат спидометра делится на зоны (до 6), цвет каждой зоны конфигурируется. Таким образом достигается ситуационное восприятие.

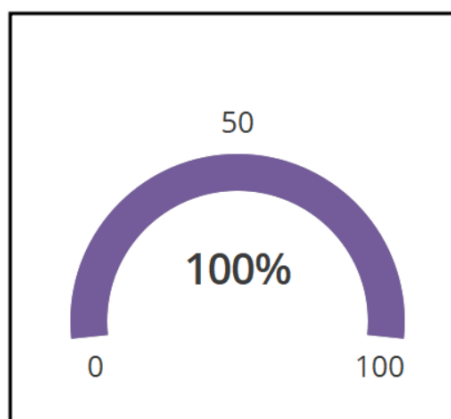
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 17 свойств.

- S Путь до объекта
- i1 Число секторов
- S Название диаграммы
- S Ед. изм.
- f4 Верхняя граница сектора 1
- f4 Верхняя граница сектора 2
- f4 Верхняя граница сектора 3
- f4 Верхняя граница сектора 4
- f4 Верхняя граница сектора 5
- u4 Цвет сектора 1
- u4 Цвет сектора 2
- u4 Цвет сектора 3
- u4 Цвет сектора 4
- u4 Цвет сектора 5
- u4 Цвет сектора 6
- u4 Цвет фоновой части
- u4 Цвет шрифта

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

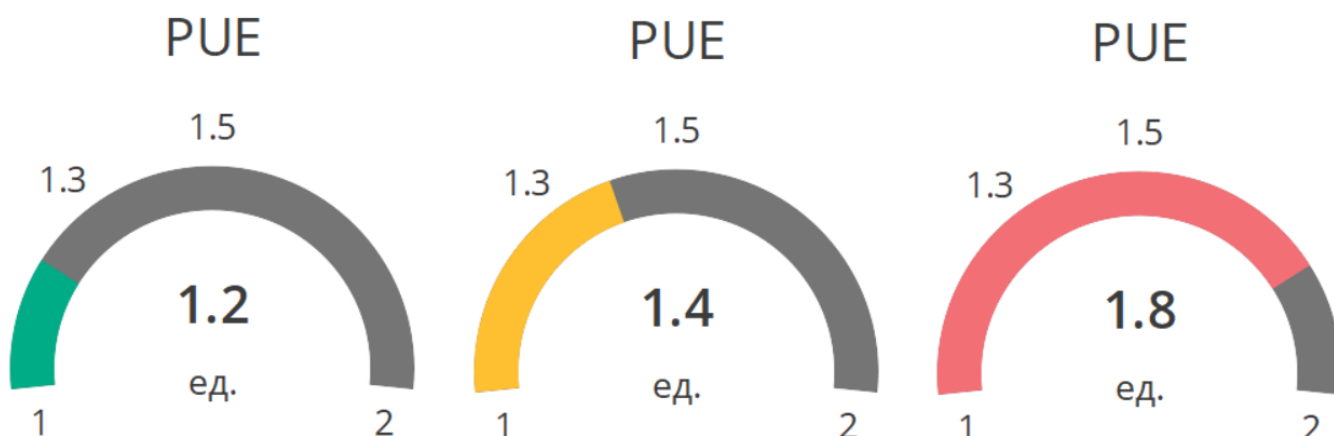
В данном примере путь до объекта: Object.RPM\_3

#### Object.RPM\_3.Value

**Число секторов.** Указывается число секторов на диаграмме. Диапазон значений: 1–6.

Примечание: текущее значение округляется до десятых долей.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом (3 примера для 3-х различных значений):



Так на диаграмме 3 сектора с заданными цветами: зеленый, желтый, красный. Если значение попадает в первый сектор, то индикатор окрашивается в соответствующий цвет. Определить попадание в сектор можно по нижеприведенному неравенству:

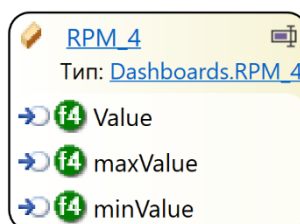
$$\text{нижняя граница} \leq x < \text{верхняя граница}$$

## 2.13. RPM\_4 (Диаграмма спидометр SA)

Диаграмма спидометр – диаграмма, отображающая уровень показателя согласно заданной шкале. Циферблат спидометра делится на зоны (до 6), цвет каждой зоны конфигурируется. Таким образом достигается ситуационное восприятие.

Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

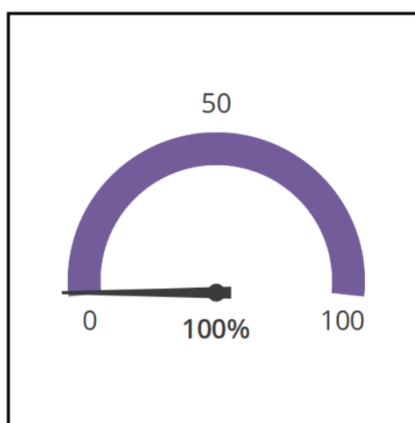
Изображение шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.



В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 17 свойств.

- S Путь до объекта
- i1 Число секторов
- S Название диаграммы
- S Ед. изм.
- f4 Верхняя граница сектора 1
- f4 Верхняя граница сектора 2
- f4 Верхняя граница сектора 3
- f4 Верхняя граница сектора 4
- f4 Верхняя граница сектора 5
- u4 Цвет сектора 1
- u4 Цвет сектора 2
- u4 Цвет сектора 3
- u4 Цвет сектора 4
- u4 Цвет сектора 5
- u4 Цвет сектора 6
- u4 Цвет фоновой части
- u4 Цвет шрифта

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

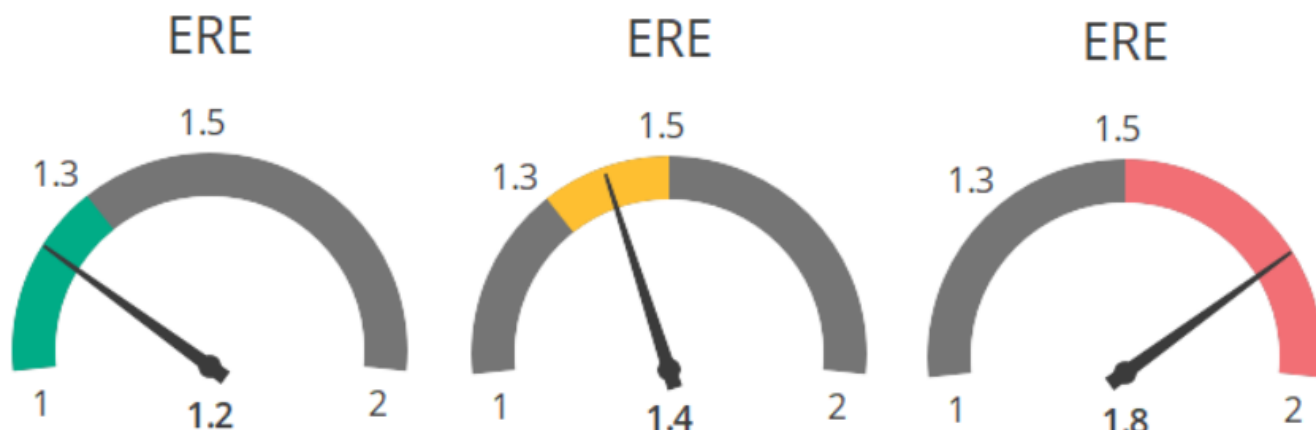
В данном примере путь до объекта: Object.RPM\_4

f4 Object.RPM\_4.Value

**Число секторов.** Указывается число секторов на диаграмме. Диапазон значений: 1–6.

Примечание: текущее значение округляется до десятых долей.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом (3 примера для 3-х различных значений):



Так на диаграмме 3 сектора с заданными цветами: зеленый, желтый, красный. Если значение попадает в первый сектор, то индикатор окрашивается в соответствующий цвет. Определить попадание в сектор можно по нижеприведенному неравенству:

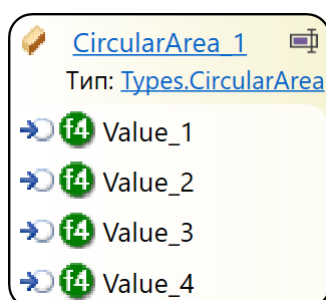
$$\text{нижняя граница} \leq x < \text{верхняя граница}$$

## 2.14. CircularArea\_1 (Лепестковая диаграмма)

Лепестковая диаграмма, также известная как паук, позволяет наглядно сравнить несколько объектов по различным критериям. Позволяет представлять данные на двухмерном графике, где каждой из переменных соответствует отдельный сегмент. Чем дальше точка находится от центра, тем выше она занимает место в категории.

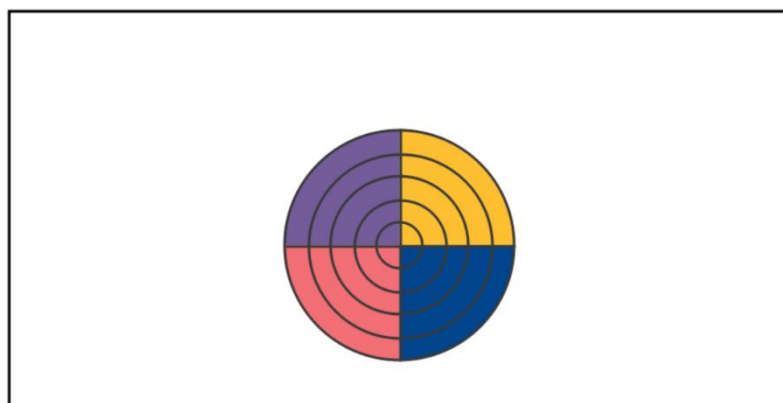
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 12 свойств.

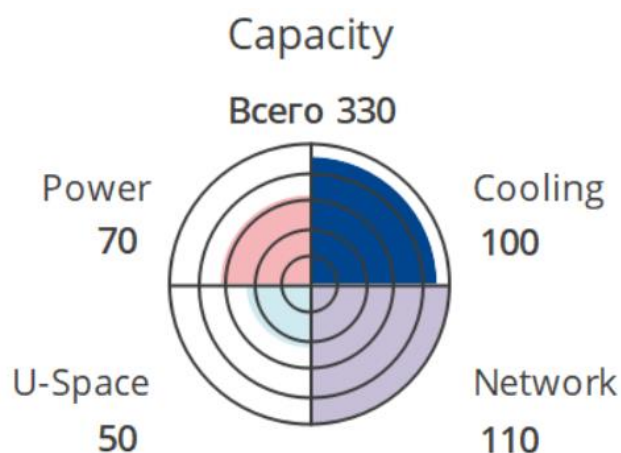
- S Путь до объекта
- S Название диаграммы
- S Подпись квадранта 1
- S Подпись квадранта 2
- S Подпись квадранта 3
- S Подпись квадранта 4
- u4 Цвет сектора 1
- u4 Цвет сектора 2
- u4 Цвет сектора 3
- u4 Цвет сектора 4
- u4 Цвет границ
- u4 Цвет шрифта

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.CircularArea\_1

f4 Object.CircularArea\_1.Value\_1

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

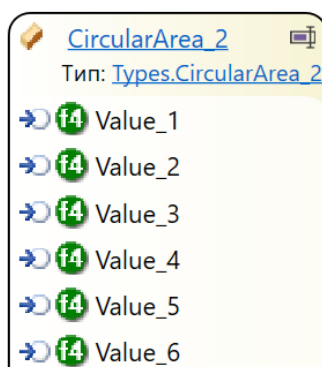


## 2.15. CircularArea\_2 (Лепестковая диаграмма)

Лепестковая диаграмма, также известная как паук, представляет собой диаграмму с двумя измерениями, на радиальных осях которой отображаются одна или несколько групп значений, для сопоставления которых используются несколько переменных. Поддерживает до 12 осей.

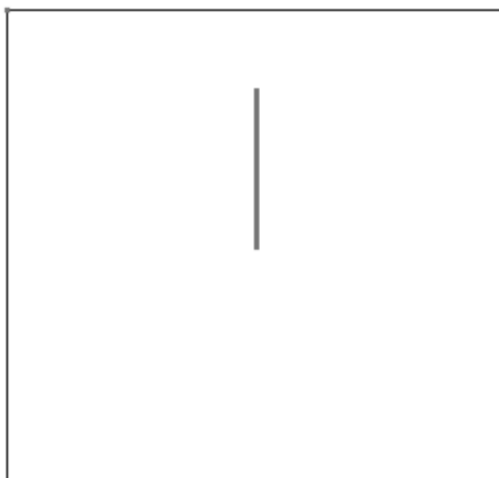
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI. Использование в Web не рекомендуется.

Изображение фрагмент шаблона в DevStudio:









На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:




Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 6 свойств.

-  Путь до объекта
-  Число осей
-  Название диаграммы
-  Цвет графика
-  Цвет границ
-  Цвет шрифта

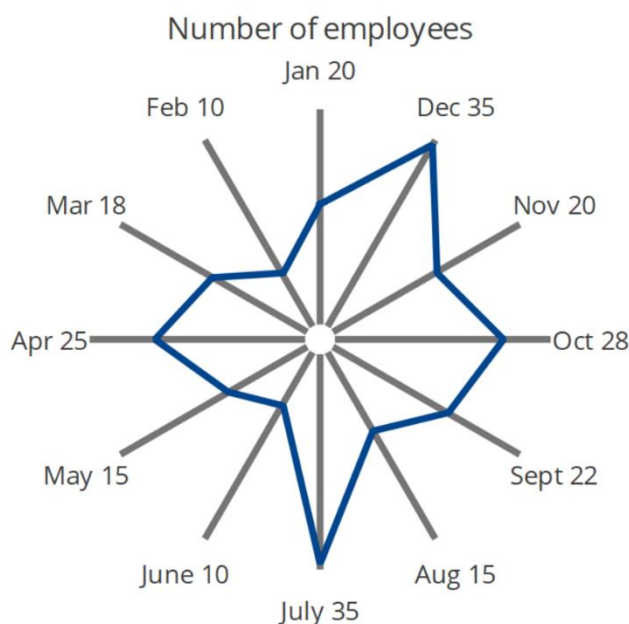
**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.CircularArea\_2

 Object.CircularArea\_2.Value\_1

**Число осей.** Указывается число осей на диаграмме. Диапазон значений: 2–12.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

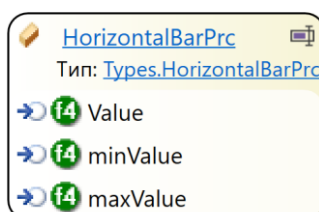


## 2.16. HorizontalBarPrc (Индикатор прогресса)

Горизонтальный индикатор прогресса – графический элемент, показывающий статус процесса с помощью заполнения прямой по мере его выполнения.

Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмент шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 5 свойств.

- S** Путь до объекта
- S** Ед. изм.
- u4** Цвет индикатора
- u4** Цвет фоновой части
- u4** Цвет шрифта

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.HorizontalBarPrc

**f4** Object.HorizontalBarPrc.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

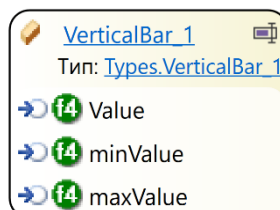


## 2.17. VerticalBar\_1 (Вертикальный индикатор прогресса)

Вертикальный индикатор прогресса – графический элемент, показывающий статус процесса с помощью заполнения вертикальной прямой по мере его выполнения.

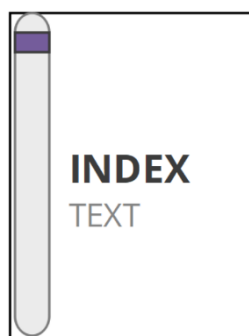
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмент шаблона в DevStudio:









На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:




Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 6 свойств.

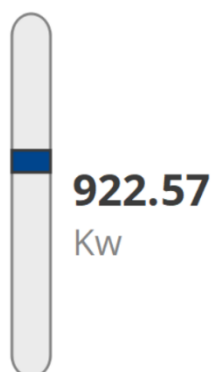
-  Путь до объекта
-  Ед. изм.
-  Цвет индикатора
-  Цвет фоновой части
-  Цвет шрифта индекса
-  Цвет шрифта ед. изм.

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.VerticalBar\_1

 Object.VerticalBar\_1.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

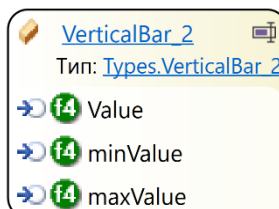


## 2.18. VerticalBar\_2 (Вертикальный индикатор прогресса)

Вертикальный индикатор прогресса – графический элемент, показывающий статус процесса с помощью заполнения вертикальной прямой по мере его выполнения.

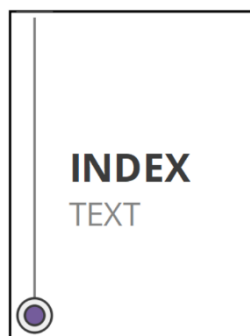
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмент шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 6 свойств.

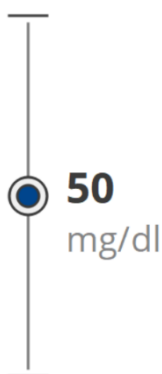
- S Путь до объекта
- S Ед. изм.
- u-1 Цвет индикатора
- u-1 Цвет шкалы
- u-1 Цвет шрифта индекса
- u-1 Цвет шрифта ед. изм.

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.VerticalBar\_2

f4 Object.VerticalBar\_2.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:



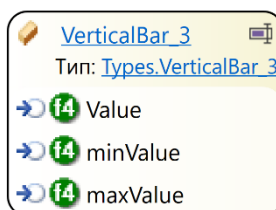
## 2.19. VerticalBar\_3 (Вертикальный индикатор прогресса)

Вертикальный индикатор прогресса – графический элемент, показывающий статус процесса с помощью заполнения вертикальной прямой по мере его выполнения.

Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.



Изображение фрагмент шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 6 свойств.

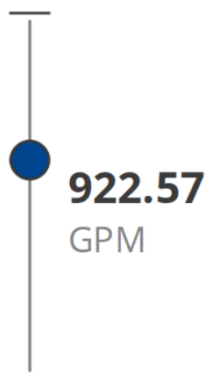
- S Путь до объекта
- S Ед. изм.
- u-1 Цвет индикатора
- u-1 Цвет шкалы
- u-1 Цвет шрифта индекса
- u-1 Цвет шрифта ед. изм.

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.VerticalBar\_3

f4 Object.VerticalBar\_3.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

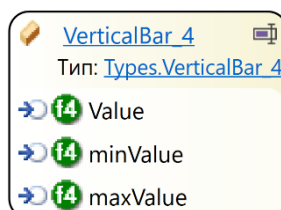


## 2.20. VerticalBar\_4 (Вертикальный индикатор прогресса)

Вертикальный индикатор прогресса – графический элемент, показывающий статус процесса с помощью заполнения вертикальной прямой по мере его выполнения.

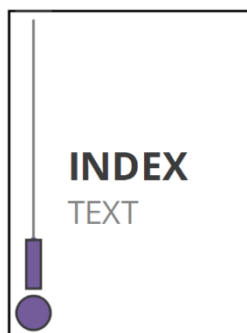
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмент шаблона в DevStudio:



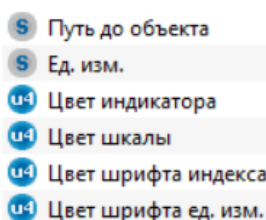
На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 6 свойств.

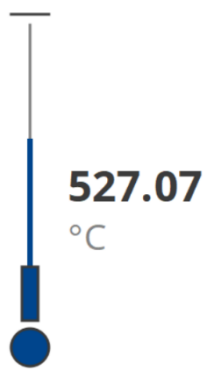


**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.VerticalBar\_4

 Object.VerticalBar\_4.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:



## 2.21. LineChart (Линейный график)

Линейный график показывает динамику по одному показателю. Для отображения исторических данных должна быть настроена база истории.

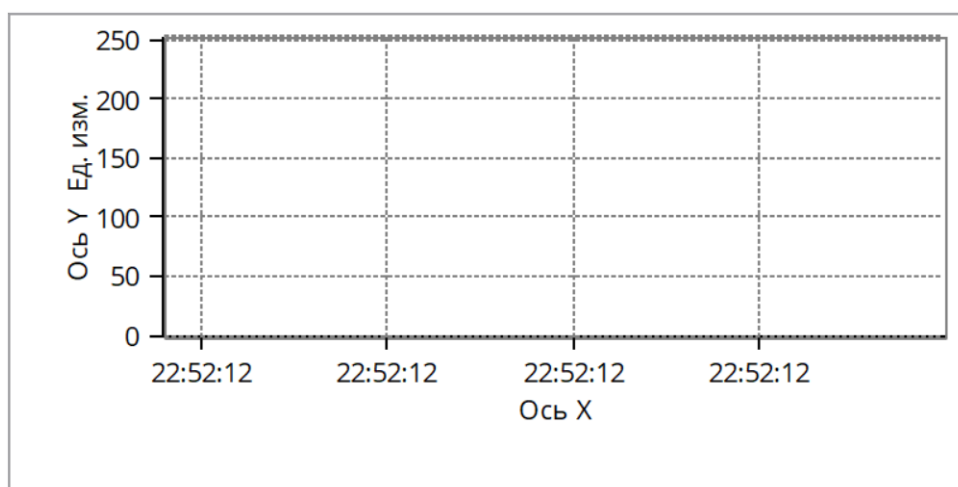
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI. Использование в Web не рекомендуется.

Изображение фрагмент шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Размер графика можно сконфигурировать в соответствии с требованиями пользователя. Также график имеет свои базовые свойства, доступные для конфигурирования. К ним относятся: цвет фона, таймлайн, легенда, границы. Также необходимо сконфигурировать 7 дополнительных свойств:

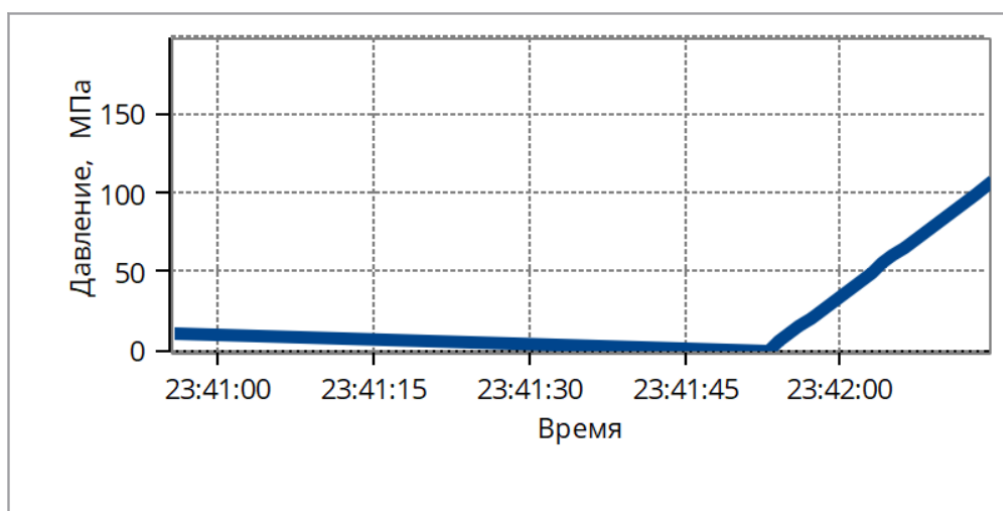
- S** Путь до объекта
- S** Название оси X
- S** Название оси Y
- S** Ед. изм. ось Y
- u4** Цвет оси X
- u4** Цвет оси Y
- u4** Цвет графика

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.LineChart

**f4** Object.LineChart.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

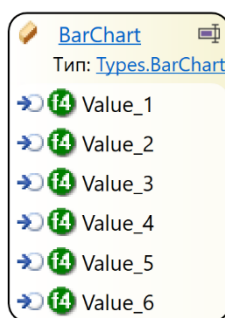


## 2.22. BarChart (Столбчатая диаграмма)

Столбчатая диаграмма отображает значения одного или нескольких показателей по каждой категории с помощью горизонтальных столбиков. Длина столбика определяет значение показателя: чем длиннее столбец, тем больше его значение.

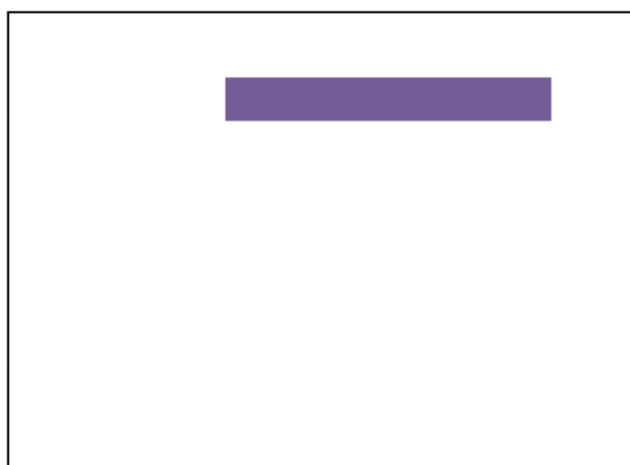
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмент шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 11 свойств.

- S Путь до объекта
- i1 Число переменных
- S Название диаграммы
- S Легенда 1
- S Легенда 2
- S Легенда 3
- S Легенда 4
- S Легенда 5
- S Легенда 6
- u-1 Цвет столбцов
- u-1 Цвет шрифта

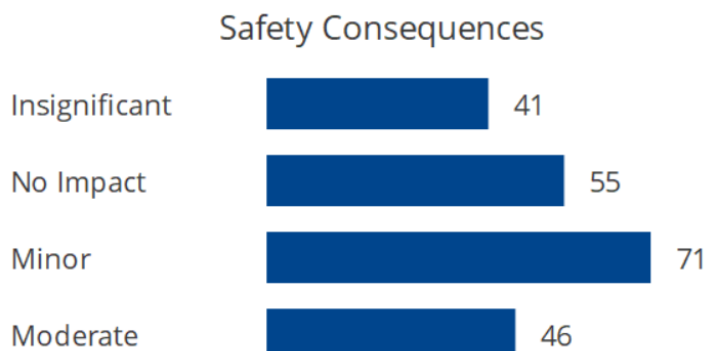
**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.BarChart

f4 Object.BarChart.Value\_1

**Число переменных.** Указывается число переменных для отображения на диаграмме. Диапазон значений: 1–6.

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

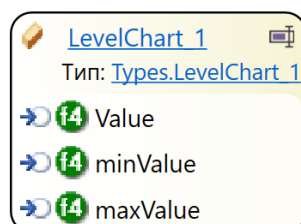


## 2.23. LevelChart\_1 (Горизонтальный уровень)

Горизонтальный уровень – графический элемент, показывающий статус процесса с помощью заполнения горизонтальной прямой по мере его выполнения.

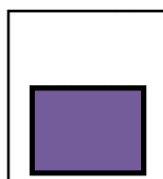
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмент шаблона в DevStudio:



На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

В режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 8 свойств.

- S Путь до объекта
- S Название диаграммы
- S Ед. изм.
- u-4 Цвет индикатора
- u-4 Цвет фоновой части
- u-4 Цвет границы
- u-4 Цвет шрифта
- u-4 Цвет шрифта за диаграммой

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.LevelChart\_1

#### Object.LevelChart\_1.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

Diesel Level For  
Generators

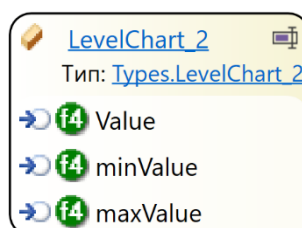


## 2.24. LevelChart\_2 (Вертикальный уровень)

Вертикальный уровень – графический элемент, показывающий статус процесса с помощью заполнения вертикальной прямой по мере его выполнения.

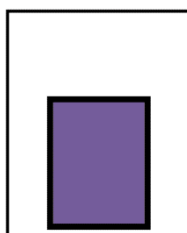
Диаграмма состоит из двух элементов: шаблон DevStudio и шаблон HMI.

Изображение фрагмент шаблона в DevStudio:











На вход каждой переменной нужно подать соответствующие данные. Если тип данных не совпадает, требуется произвести конвертирование типов.

режиме дизайнера в HMI диаграмма выглядит следующим образом:



Прямоугольная рамка помогает определить границы объекта в режиме разработки.

Для конфигурации диаграммы необходимо указать 8 свойств.

-  Путь до объекта
-  Название диаграммы
-  Ед. изм.
-  Цвет индикатора
-  Цвет фоновой части
-  Цвет границы
-  Цвет шрифта
-  Цвет шрифта за диаграммой

**Путь до объекта.** Путь до объекта (включая сам объект) можно посмотреть через OPCExplorer, выбрав любой сигнал. Путь указывается без точки. В свойстве не требуется указывать полный тэг до сигнала.

В данном примере путь до объекта: Object.LevelChart\_2

 Object.LevelChart\_2.Value

После конфигурирования объекта и запуска в режиме Runtime рамка пропадет. Диаграмма будет выглядеть следующим образом:

Diesel Level For  
Generators







Узнать обо всех продуктах  
Systeme Soft

<https://systemesoft.ru/>

